

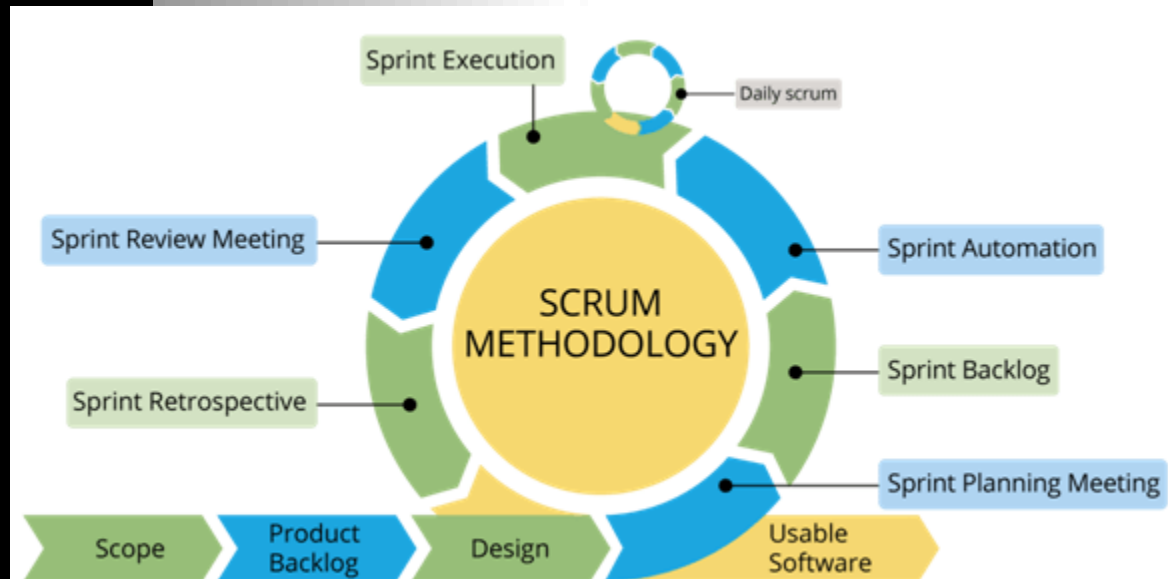
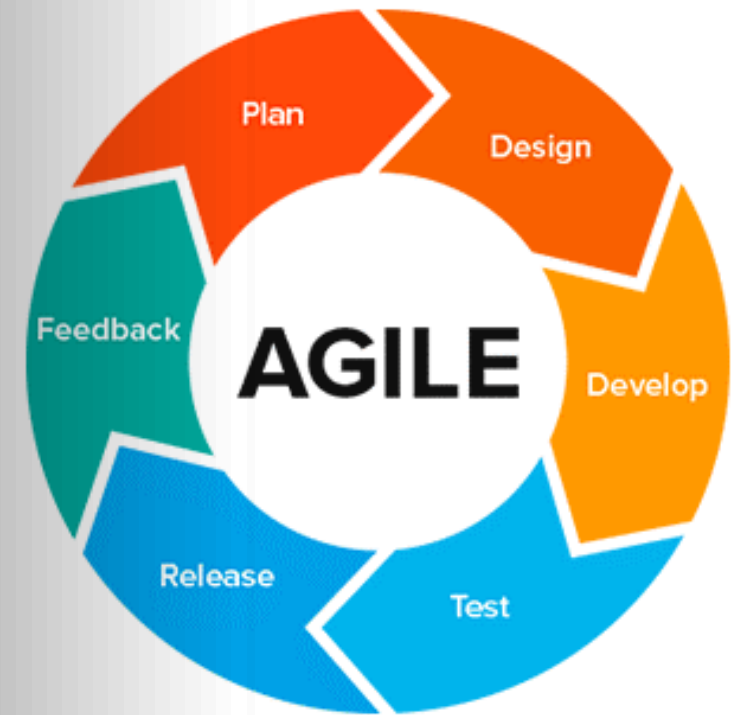


ToDoApi

By Nijo John



Introduction

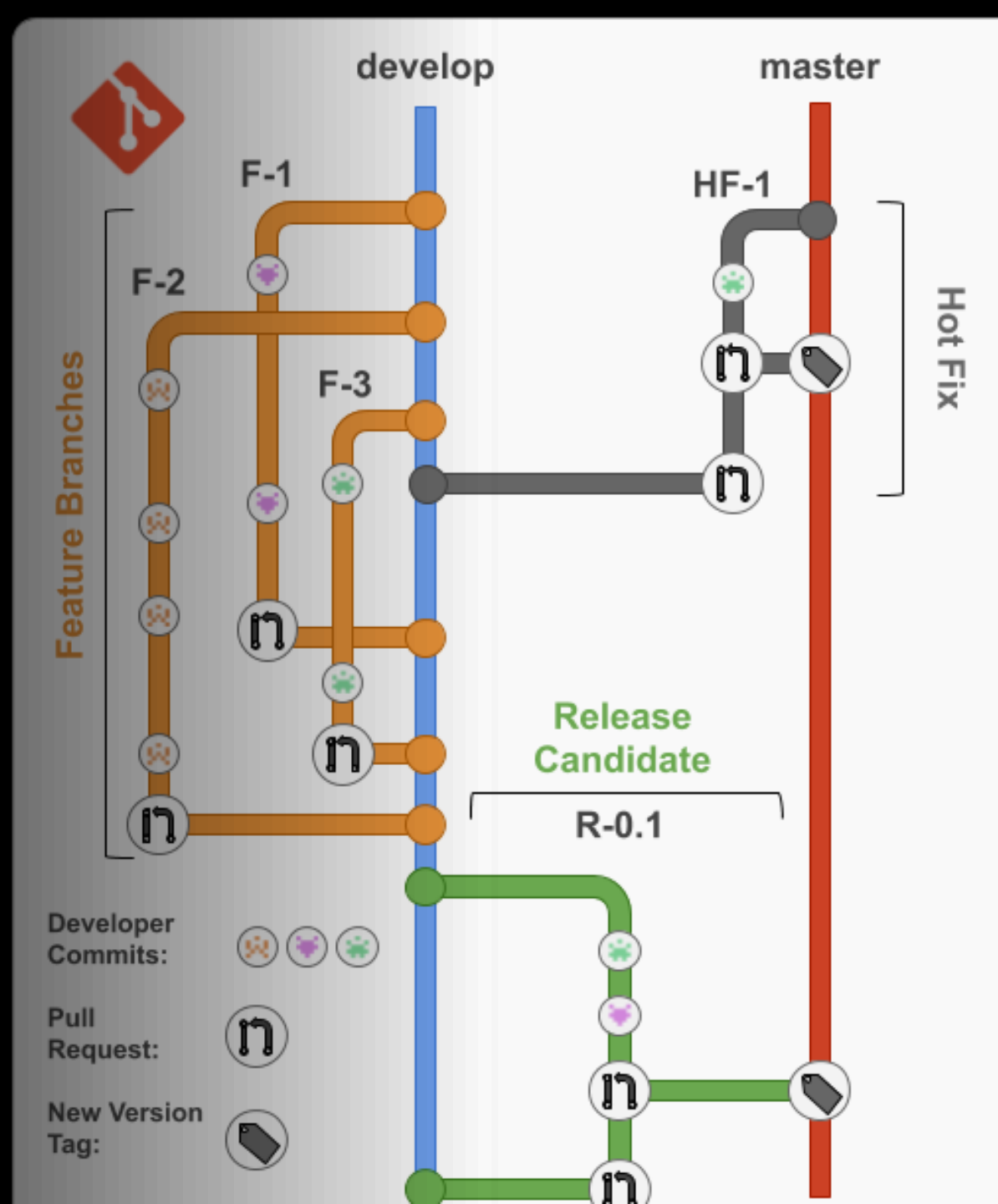


Journey

- Java (OOP)
- Maven
- GCP
- Jira
- Github
- Springboot
- HTML
- CSS
- Javascript

CI

Version controlling
(Master/dev/features)



Testing

The screenshot displays an IDE interface with three main panels. The top panel shows the source code of `ItemControllerUnitTest.java`, which is a Spring Boot test class. It includes imports for `org.assertj.core.api.Assertions` and `org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc`. The class is annotated with `@SpringBootTest` and `@ActiveProfiles("dev")`. It contains several test methods: `createTest()` which sets up test data and a mock service, `whenCreateTest()` which tests the `create` method of the `ItemController`, and `readOne()` which tests the `readOne` method. The bottom-left panel shows the 'JUnit' test runner output, indicating that the tests passed successfully after 14.568 seconds. The bottom-right panel shows the 'Coverage' report for the `ItemControllerUnitTest` class, detailing the percentage of code covered by the tests.

```
1 package com.qa.shopping.rest;
2
3 import static org.assertj.core.api.Assertions.assertThat;
4
5 @SpringBootTest // spring boot test lets spring know this is a test file and treat as such
6 @ActiveProfiles("dev") // lets us set the application properties file.
7 public class ItemControllerUnitTest {
8     @Autowired
9     private ItemController controller;
10
11     @MockBean
12     private ItemService service;
13
14     @Autowired
15     private ModelMapper mapper;
16
17     // same thing from our service
18     private ItemDto mapToDto(Item item) {
19         return this.mapper.map(item, ItemDto.class);
20     }
21
22     // During our test we want give it some data to use
23     private final Item TEST_ITEM_1 = new Item(1L, "PS4", "Console", 1);
24     private final Item TEST_ITEM_2 = new Item(2L, "PSP", "Console", 1);
25     private final Item TEST_ITEM_3 = new Item(3L, "Nintendo DS", "Console", 1);
26     private final Item TEST_ITEM_4 = new Item(4L, "Sony SRS-XB30", "Speakers", 1);
27
28     // I also want to create a list of cars that i can use later
29     private final List<Item> LISTOFITEMS = List.of(TEST_ITEM_1, TEST_ITEM_2, TEST_ITEM_3, TEST_ITEM_4);
30
31     // Create
32     @Test
33     void createTest() throws Exception {
34         when(this.service.create(TEST_ITEM_1)).thenReturn(this.mapToDto(TEST_ITEM_1));
35         assertThat(new ResponseEntity<ItemDto>(this.mapToDto(TEST_ITEM_1), HttpStatus.CREATED))
36             .isEqualTo(this.controller.create(TEST_ITEM_1));
37         verify(this.service, atLeastOnce()).create(TEST_ITEM_1);
38     }
39
40     // Read one
41     @Test
42     void readOne() throws Exception {
43         when(this.service.readOne(1L)).thenReturn(this.mapToDto(TEST_ITEM_1));
44         assertThat(this.controller.readOne(1L))
45             .isEqualTo(this.mapToDto(TEST_ITEM_1));
46         verify(this.service, atLeastOnce()).readOne(1L);
47     }
48 }
```

JUnit Test Results:

- Finished after 14.568 seconds
- Runs: 7/7
- Errors: 0
- Failures: 0

ItemControllerUnitTest [Runner: JUnit 5] [0.275 s]

Failure Trace

Boot Dashboard

local

- CarProject
- LibraryProject
- ShoppingToDo

ItemControllerUnitTest (18 Dec 2020 11:44:53)

Element	Coverage	ed Instructions	ed Instructions	tal Instructions
ShoppingToDo	19.0 %	553	2,359	2,912
src/test/java	21.1 %	346	1,292	1,638
com.qa.shopping.rest	28.7 %	346	861	1,207
com.qa.shopping.service	0.0 %	0	427	427
com.qa.shopping	0.0 %	0	4	4
src/main/java	16.2 %	207	1,067	1,274



Demo

TODO_WebApi

Classic software project

Back to project

Filters

All issues

My open issues

Reported by me

Open issues

Done issues

Viewed recently

Resolved recently

Updated recently

View all filters

Projects / TODO_WebApi

Issues

Export Issues

Go to advanced search

Search issues

Assignee

Reporter

Status: Done

Type

Clear all

Switch to detail view

Type	Key	Summary	Assignee	Reporter	P	S
<input checked="" type="checkbox"/>	API-31	Upload a copy of presentation in PDF format to git	Unassigned	Nijo John	↑	DC
<input checked="" type="checkbox"/>	API-30	Create Presentation to present	Unassigned	Nijo John	↑	DC
<input checked="" type="checkbox"/>	API-29	SonarQube for static analysis	Unassigned	Nijo John	↑	DC
	API-28	Static Analysis	Unassigned	Nijo John	↑	DC
<input checked="" type="checkbox"/>	API-27	Spring development	Unassigned	Nijo John	↑	DC
	API-26	Backend Development	Unassigned	Nijo John	↑	DC
<input checked="" type="checkbox"/>	API-25	JavaScript Development	Unassigned	Nijo John	↑	DC
<input checked="" type="checkbox"/>	API-23	HTML development	Unassigned	Nijo John	↑	DC
	API-22	Front-End Development	Unassigned	Nijo John	↑	DC
<input checked="" type="checkbox"/>	API-21	UML diagram	Unassigned	Nijo John	↑	DC

Give feedback

Showing 1-28 of 28

< 1 >

Sprint review

Sprint Retrospective



What went well?



What could've improved?



Conclusion

A large, stylized graphic on the right side of the slide. It features the words "The End" in a white, cursive script font with a dark blue shadow. The text is centered over a dark blue circle, which is itself centered within a series of concentric red circles that create a tunnel-like effect. The background of the entire slide is black.